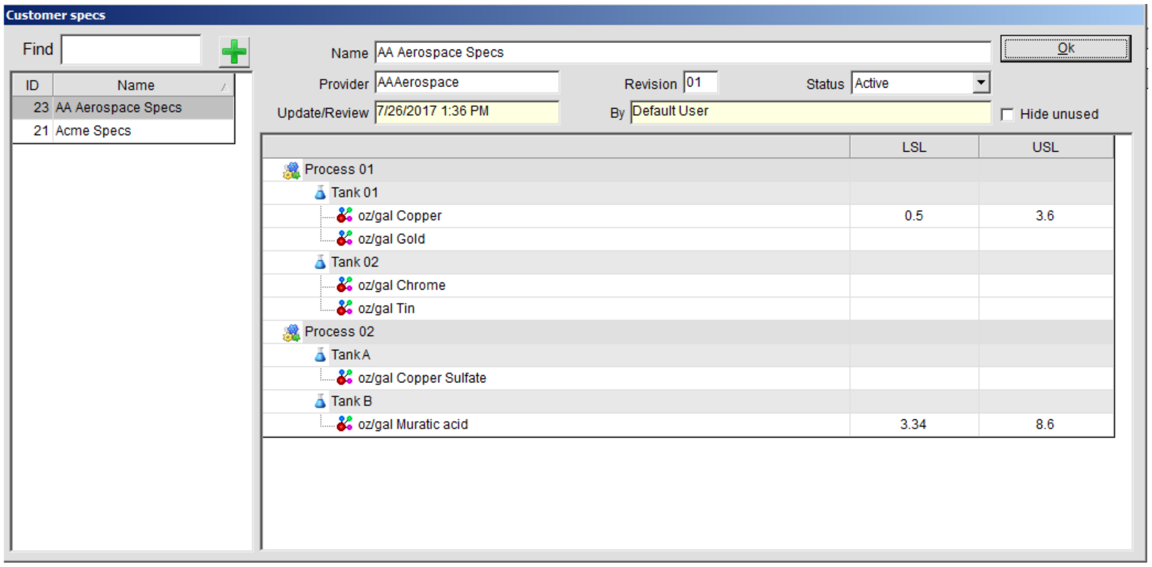
TrueLogic's: Customer Specs

The Customer Spec system allows for the creation of groups of specs that are related in some way. Often, they are from the same company. For example, you may have a customer spec called "ABC Aerospace Specs" and in that customer spec would be lower and upper specs for one or more result items. You may have another customer spec called, say, "XYZ Company Specs".

Note that the customer spec system is not turned on by default. If you'd like this feature turned on, please contact TrueLogic Customer Support.

Managing the Customer Specs

To manage your customer specs, click Tools > Manage > Customer Specs. From this screen you can create a new customer spec by clicking the green "+" button (see Figure 1). Double click a cell in the LSL or USL column to enter a spec value. If you wish to temporarily deactivate a customer spec (so it doesn't show in the graphs or cause a rule action to be taken) then change the Status to Inactive.



ID	Name
23	AA Aerospace Specs
21	Acme Specs

Name: AA Aerospace Specs
Provider: AAAerospace
Revision: 01
Status: Active
Update/Review: 7/26/2017 1:36 PM
By: Default User
☐ Hide unused

	LSL	USL
Process 01		
Tank 01		
oz/gal Copper	0.5	3.6
oz/gal Gold		
Tank 02		
oz/gal Chrome		
oz/gal Tin		
Process 02		
Tank A		
oz/gal Copper Sulfate		
Tank B		
oz/gal Muratic acid	3.34	8.6

Figure 1: Managing Customer Specs

Customer Specs In The Graphs

If the "Show Customer Specs" option is checked in the Graph options, and a customer spec's status is set to "Active" (see above) then its specs will show on the Run Chart (see Figure 2). Customer specs are labeled on the right side of the graph as "CSU" (Customer Spec Upper) or "CSL" (Customer Spec Lower). Clicking the mouse on those labels displays information about that customer spec.

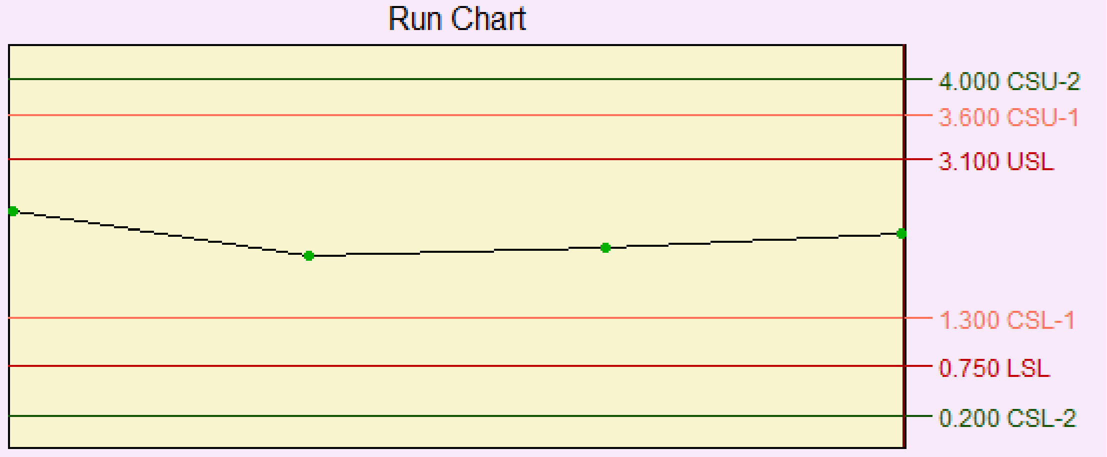


Figure 2: Customer Specs in a Run Chart

TrueChem Variables for Customer Specs

[rst.CustSpecState] inspects all the Active customer specs that have specs for the result. It returns 0 for pass, 1 for Failed Lower, 2 for Failed Upper, 3 for Failed Both. This variable will often be used in a Rule. If no parameters are used with this variable then it inspects only Active customer specs, else it inspects Active and Inactive customer specs.
Examples:

[rst.CustSpecState] causes it to look at all the Active customer specs.
[rst.CustSpecState(Provider=AA Aerospace, Status=Inactive)] causes it to look at only the customer specs for the AA Aerospace provider that are also Inactive.
[rst.CustSpecState(Name=AA Aerospace Specs)] causes it to look at only the customer spec called AA Aerospace Specs. Since a parameter is being used, that customer spec may be Active or Inactive.

[rst.CustSpecPass] or **[rst.CustSpecFail]**
Returns a list of customer specs that the result passed (or failed). This information is returned in multiple rows consisting of 4 fields (Provider name, Customer Spec name, Lower Spec, Upper Spec).
If no parameters are used with this variable then it inspects only Active customer specs, else it will inspect Active and Inactive customer specs.
This variable can have parameters of the form: field1=value1, field2=value2
Examples:

[rst.CustSpecPass] returns the customer spec information for each Active customer spec that the result passed.
[rst.CustSpecFail] returns the customer spec information for each Active customer spec that the result failed.
[rst.CustSpecPass(Provider=AA Aerospace)] returns customer spec information for just the AA Aerospace customer specs that the result

passed. Note that since a parameter is being used, it can return Inactive customer specs. To omit those, use Status=Active as a parameter.
[rst.CustSpecPass(ID=123)] returns customer spec information for just the customer spec whose ID is 123 and the result must have passed it. Note that since a parameter is being used, it can return Inactive customer specs. To omit those, use Status=Active as a parameter.

[rst.CustSpecList]
Returns information about customer specs that the result passed or failed (depending on "State" in the parameter list). The parameters can be just field names, or of the form field=value. The field=value designates conditions that must be met. You must set the State equal to Fail or Pass. The field names by themselves designate what fields to return (and their order). The returned fields will have 3 spaces between them. If no field names are listed by themselves, then this variable returns Provider, Customer Spec Name, Lower Spec, Upper Spec.
Examples:

[rst.CustSpecList(State=Fail, Status=Active, Name, Provider, Lower)] returns the customer spec Name and then 3 spaces and then the Provider and then 3 spaces and then the lower spec. And it only looks at customer specs that the Result failed either the upper or the lower spec, and that are Active.
[rst.CustSpecList(State=2, Provider=AA Aerospace)] returns customer spec information for just the AA Aerospace customer specs where the Result failed the upper spec (State=2). Because Status wasn't used in a condition, it will return both Active and Inactive customer specs. Since no fields are listed by themselves, the information returned will be Provider, Customer Spec Name, Lower Spec, Upper Spec

Notice that when using State in a field=value condition, you can specify that State=Fail, or that it equals Pass. To be more specific, you can set it equal to 0 (meaning pass), 1 (fail lower), 2 (fail upper).

Using the Customer Spec variables in Rules

Most often, a rule will simply state **If [rst.CustSpecState] > 0 Then** do a rule action. That is, if the Result broke any active Customer Specs then the rule action will get executed.
The rule action might display a message or send an email. Both can display a list of the active Customer Specs that the result failed by simply using the **[rst.CustSpecFail]** variable with no parameters.